LA-UR-21-24894
Approved for public release; distribution is unlimited.

Title: Accessing Microsoft Access Databases Using ODBC and RODBC

Author(s): Collins, David H. Jr.

Intended for: Report

Issued: 2021-05-20

Accessing Microsoft Access™ Databases Using ODBC and RODBC

ODBC (Open Database Connectivity) is an industry-standard API (Application Program Interface) that provides a standard interface, based on SQL (Structured Query Language) between applications and databases. This insulates applications from specific details of different database management systems (DBMS). Microsoft Windows™ provides an implementation of ODBC, which, along with drivers for various databases, supports the API. Windows also provides a driver for Access databases.

RODBC is a package for the R statistical programming language that provides an interface between R programs and the Windows ODBC manager. Figure 1 shows an overview of the relationship between an R program, RODBC, various Windows components, and an Access database.

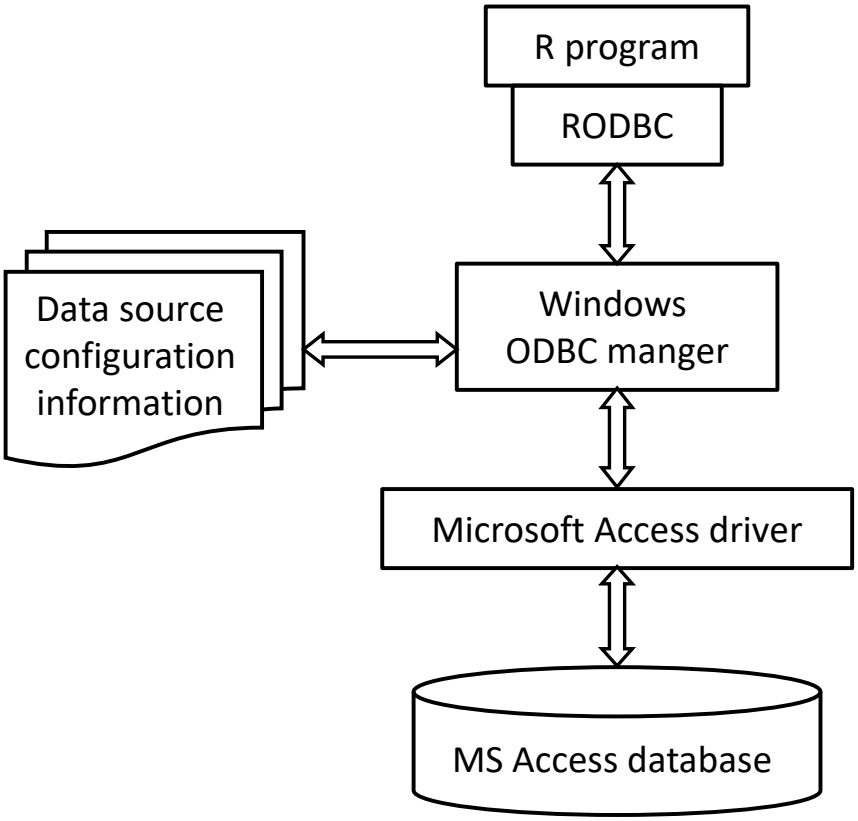

Figure 1: Overview of ODBC and RODBC under Windows

A very brief tutorial on Access

As an introduction to Access, create a sample database, to be used with the R program below, with these steps:

1. Create a new directory (folder) for testing, named `Test_Access`.[1]
2. Copy the following lines into a text file, exactly as is (do not insert any spaces or tabs):

   ```
   Name  Location      EmpNo      Status
   Dave  TA-3   1234   Active
   Joe   TA-43  4028   Active
   Zelda TA-3   2346   Active
   Gene  TA-84  6073   Retired
   Susan TA-3   7809   Leave
   ```

   Insure there are no blank lines at the end of the file, and save it as `emp.txt` in `Test_Access`.
3. Type `access` into the search box at the lower left of the screen, and open the Microsoft Access application. Click on the template `Blank desktop database`. In the dialog that pops up, name the database `emp.accdb`, select `Test_Access` as the location to save it, and click `Create`.
4. Select the `External Data` menu, then click on the `Text File` icon (the one right above `XML File`). Browse to `emp.txt` as the file to be imported and click `OK`. In the dialog that pops up select `Delimited` and click `Next`. Check the box that says `First Row Contains Field Names,` then click `Finish`.
5. (Optional) Double-click `Tables/Emp` in the Access window to verify the data were entered correctly. Close the access window.

The next step is to define the database to the Windows ODBC manager.[2] In a Windows command prompt (if you don't see this on the taskbar, type `cmd.exe` in the search box), enter `\Windows\SysWOW64\odbcad32` to invoke the dialog shown in Figure 2 (you do *not* need to be in administrator mode to do this[3]). The title should say "32 bit". There will probably already be an entry for `MS Access Database` (added when Access was installed). If so, click `Configure`. If not, click `Add` and select from the list `Microsoft Access Driver (*.mdb, *.accb)`, then click `Finish`. In either case, you will see the configuration dialog, shown in figure 3. Enter `Employee Database` for the data source name (this is the name you will use to connect to the database from R). Under Database click `Select`, select

[1] Computer inputs and outputs are shown in Courier font.
[2] Though not recommended, an alternative to this procedure is to use fully qualified connection strings in the R program. See the appendix, "Connecting using connection strings".
[3] You will be entering a `User DSN`, which will be accessible only to one user. Entering the same information as a `System DSN` makes it available to all users, and *does* require administrator privileges.

`Test_Access\emp.accb`, and click `OK`. Click `OK` in the configuration dialog; click `OK` in the administrator dialog.

Now create a second table in the `emp.accdb` database file by saving the following text as `travel.txt`, and importing it as described above. There is no need to do any additional ODBC configuration.

EmpNo	Date	Destination	Conference	Cost
1234	Chicago	7/31/2020	JSM	2000
1234	New Orleans	2/3/2010	ACM	1000
7809	San Francisco	5/10/2015	CSP	3500
4028	Seattle	8/1/2021	JSM	2400

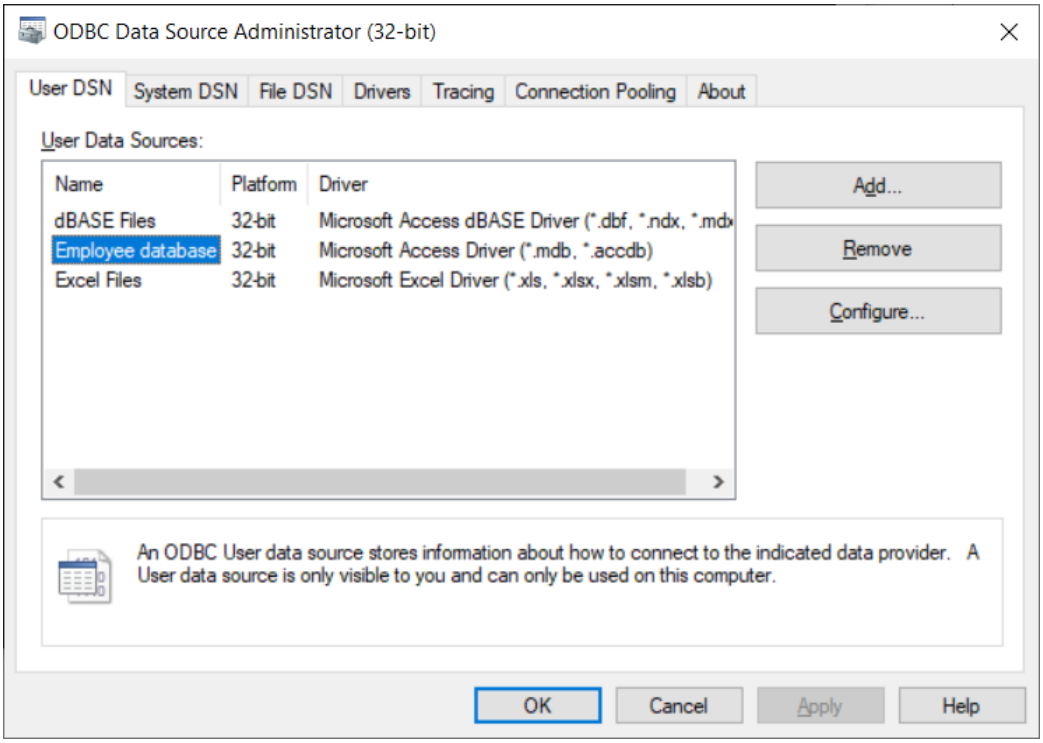

Figure 2: Windows ODBC Data Source Administrator dialog

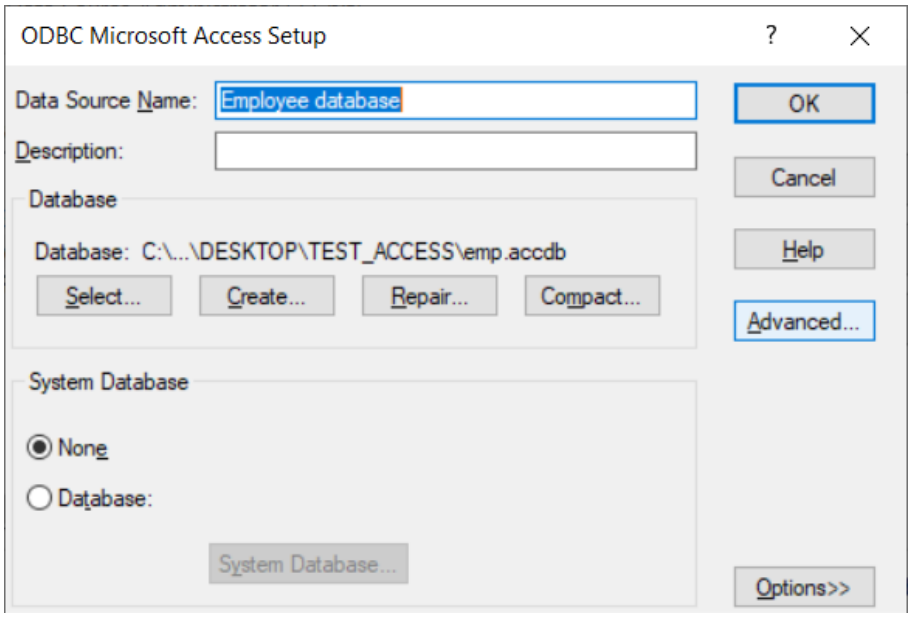

Figure 3: Windows ODBC Data Source configuration dialog

Accessing the database from R

First, since the ODBC driver for Access is 32 bit, you must run 32 bit R. Attempting to connect to a database from 64 bit R will result in an error: `The specified DSN contains an architecture mismatch between the Driver and Application`. When opening R, the initial console messages will indicate which version is being run. The name of the RGui shortcut for 32-bit R normally starts with `R i386`; to change versions in RStudio use the menu `Tools/Global options`, click on `R version/Change`, then select `Use your machine's default version of R (32 bit)`.

The sample R program below accesses the Employee Database and performs some simple queries to illustrate the use of SQL though R. To learn more about the SQL language, there are numerous tutorials on the Worldwide Web, for example
`https://www.w3schools.com/sql/`
which is also useful as a reference.

Some of the statements in the sample program are only intended to illustrate the capabilities of RODBC, and are not needed in a typical program. Executing the program line by line will provide examples of the capabilities of RODBC.

One thing to note is that reading in entire tables and processing them with program logic is generally less efficient, and often less clear, than putting the logic into SQL queries; for example, see the query commented "Which employees traveled to JSM?" in the sample program.

Sample program

```
library(RODBC)
# List the available ODBC data sources (drivers)
odbcDataSources(type="all")
# Open a connection to the sample database
connection <- odbcConnect("Employee Database")
# Returns information on the database and driver
# associated with the connection.
odbcGetInfo(connection)
# Returns detailed information on datatypes, etc.,
# for each column in the sample tables.
sqlColumns(connection, "Emp")
sqlColumns(connection, "Travel")
# Get information on how the driver converts between
# R and SQL datatypes. Use setSqlTypeInfo to
# change the type mappings.
getSqlTypeInfo("ACCESS")
# Read an entire table into a data frame.
# Use sqlSave to rewrite a table from a data frame.
sqlFetch(connection, "Emp")
# Sample SQL queries
sqlQuery(connection, "select * from Emp order by EmpNo")
sqlQuery(connection,
     "select Name, EmpNo from Emp where Status = 'Retired' ")
# Which employees traveled to JSM?
sqlQuery(connection,
      paste("select Name, EmpNo, Status from Emp where EmpNo in ",
      "(select EmpNo from Travel where Conference = 'JSM')") )
# Close the database connection.
odbcClose(connection)
```

Appendix: Connecting using connection strings

As an alternative to defining the database using the ODBC administration dialog as described above, a database connection may be made by explicitly identifying the database driver and the location of the database file. For our example, the following statements replace the call to `odbcConnect` in the sample program:

```
connectString <-
   paste("Driver={Microsoft Access Driver (*.mdb, *.accdb)};",
         "Dbq=C:\\Test_Access\\emp.accdb;")
connection <- odbcDriverConnect(connectString)
```

Notice that in specifying the database file path, backslashes must be doubled, since backslash in a connection string is interpreted as an escape character.

This can be useful for testing, but is not recommended in general because it ties the program to a specific database system, release level, and file location. If the database reference is only to a data source name defined to ODBC, we could, for example, change from an Access database to the same structure in an Oracle or DB2 database, without changing the program.

Detailed information on connection string formats for many DBMS can be found at `https://www.connectionstrings.com/`